FSD Laboratory 06

Name - Shounak Dighe
Roll No - 20
Pannel - H
Prn - 1032233107

Aim: Develop a set of REST API using Express and Node.

Objectives:

1. To define HTTP GET and POST operations.
2. To understand and make use of 'REST', 'a REST endpoint', 'API Integration', and 'API Invocation'
3. To understand the use of a REST Client to make POST and GET requests to an API.

Theory:

What is a REST API?

A REST API (Representational State Transfer Application Programming Interface) is a set of rules and conventions for building and interacting with web services. It leverages HTTP methods to perform operations on resources, which are typically represented in a web-based system. REST APIs are designed to be stateless and use standard HTTP methods to request or manipulate resources.

Main Purpose of REST API

The primary purpose of a REST API is to provide a simple and standardized way for applications to communicate with each other over the web. It enables different software systems to interact using a common interface, making it easier to integrate and exchange data between disparate systems.

Key goals of a REST API include:

- **Simplicity**: It uses standard HTTP methods, which are widely understood and supported.
- **Scalability**: REST APIs are stateless, meaning each request from a client to the server must contain all the information needed to understand and process the request.
- **Interoperability**: By adhering to common standards, REST APIs allow different applications and systems to work together.
- **Performance**: It can leverage caching and other HTTP features to improve performance and reduce load.

FAQ:

What are HTTP Request Types?

HTTP request types, also known as HTTP methods, define the type of action that the client wants to perform on a resource. The main types are:

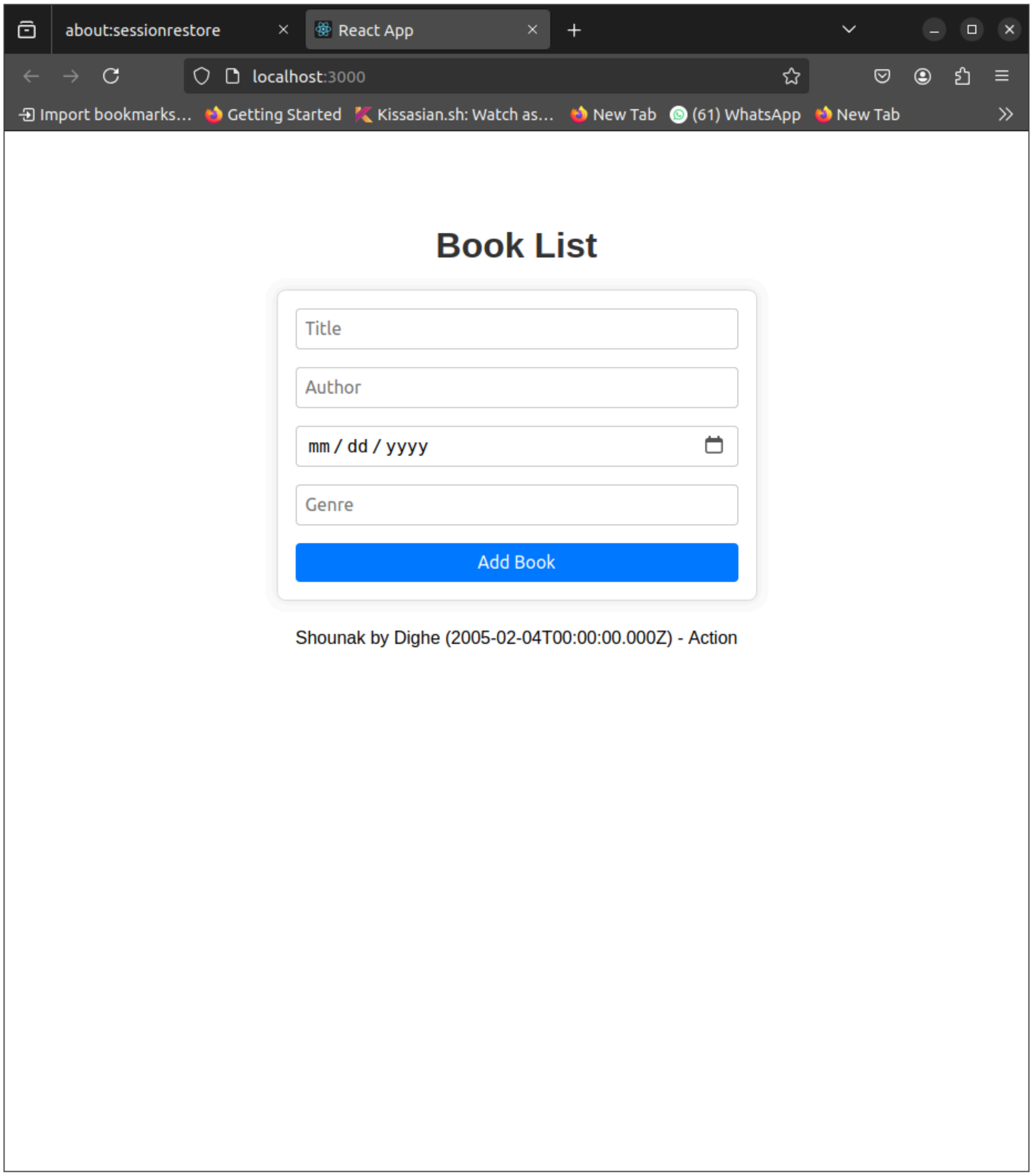
1. **GET**: Retrieves data from a server. It's used to request a resource without modifying it. For example, fetching user details from a database.
2. **POST**: Submits data to be processed to a specified resource. It's often used to create new resources or submit form data. For example, submitting a new user registration form.
3. **PUT**: Updates an existing resource with new data. If the resource does not exist, it may create a new one. For example, updating user profile information.
4. **DELETE**: Removes a specified resource from the server. For example, deleting a user account.
5. **PATCH**: Applies partial modifications to a resource. Unlike PUT, which replaces the entire resource, PATCH only changes the specified fields. For example, updating just the email address of a user.
6. **HEAD**: Retrieves the headers of a resource, similar to GET, but without the actual resource data. It's often used to check if a resource has been modified.
7. **OPTIONS**: Describes the communication options for the target resource. It's used to determine the supported methods or functionalities of a resource.
8. **TRACE**: Echoes back the received request, mainly used for diagnostic purposes to see how the request is being handled by the server.

Problem Statements:

Creating and adding new book records in the book database using REST API.

Help Link:

https://stackabuse.com/building-a-rest-api-with-node-and-express/



Book List

Title

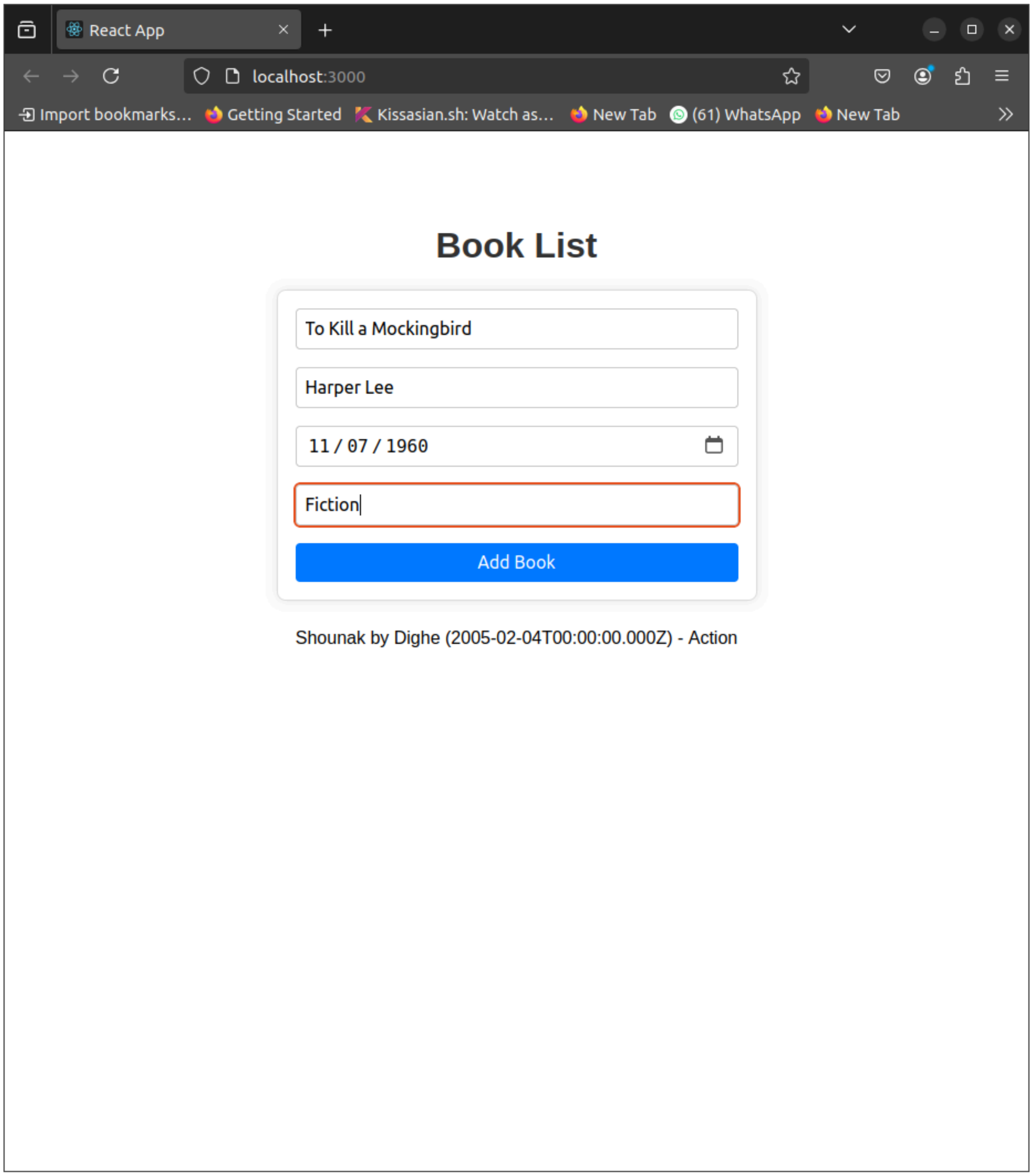
Author

mm / dd / yyyy

Genre

Add Book

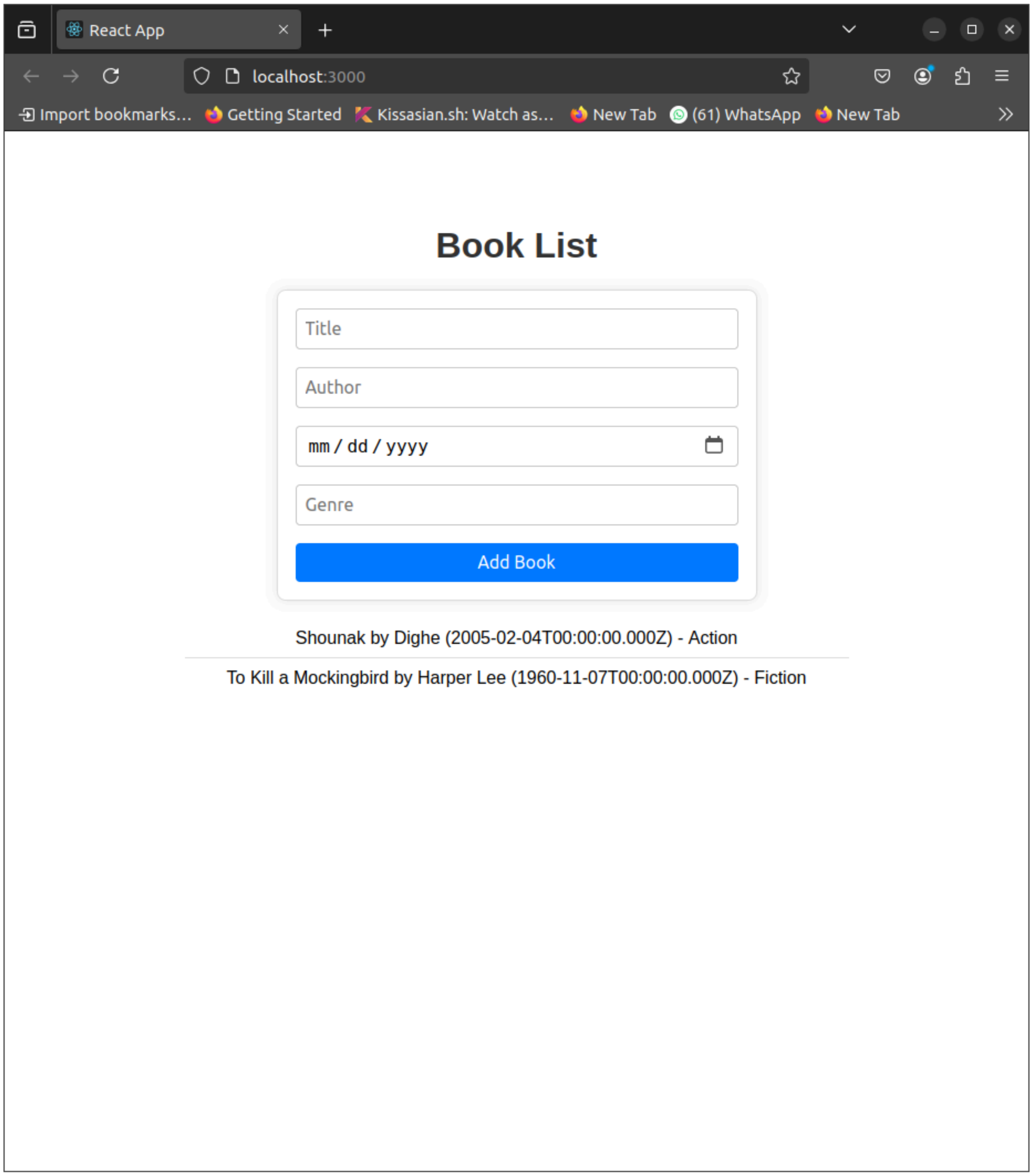
Shounak by Dighe (2005-02-04T00:00:00.000Z) - Action



Book List

To Kill a Mockingbird
Harper Lee
11 / 07 / 1960
Fiction

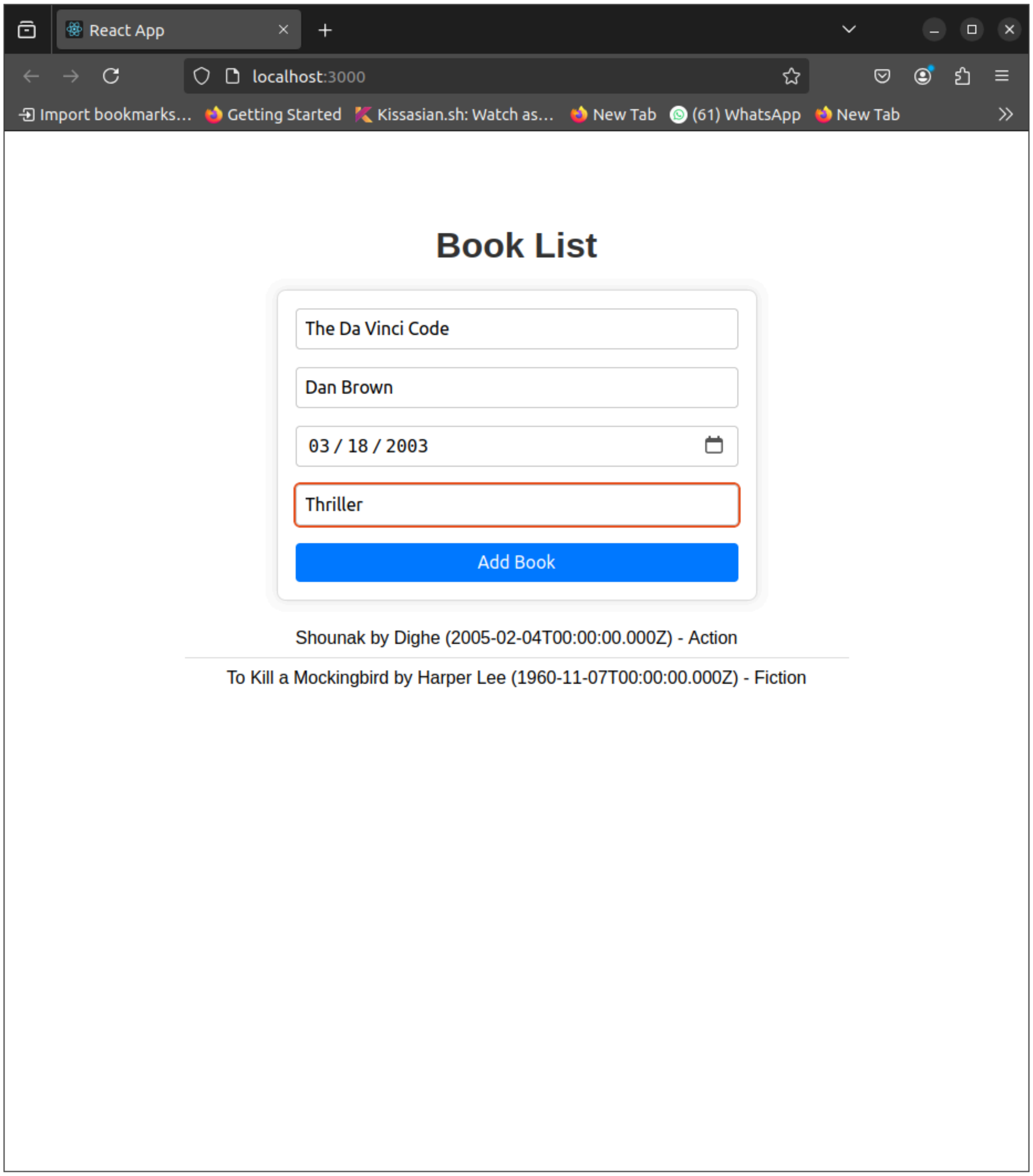
Shounak by Dighe (2005-02-04T00:00:00.000Z) - Action



Book List

Shounak by Dighe (2005-02-04T00:00:00.000Z) - Action

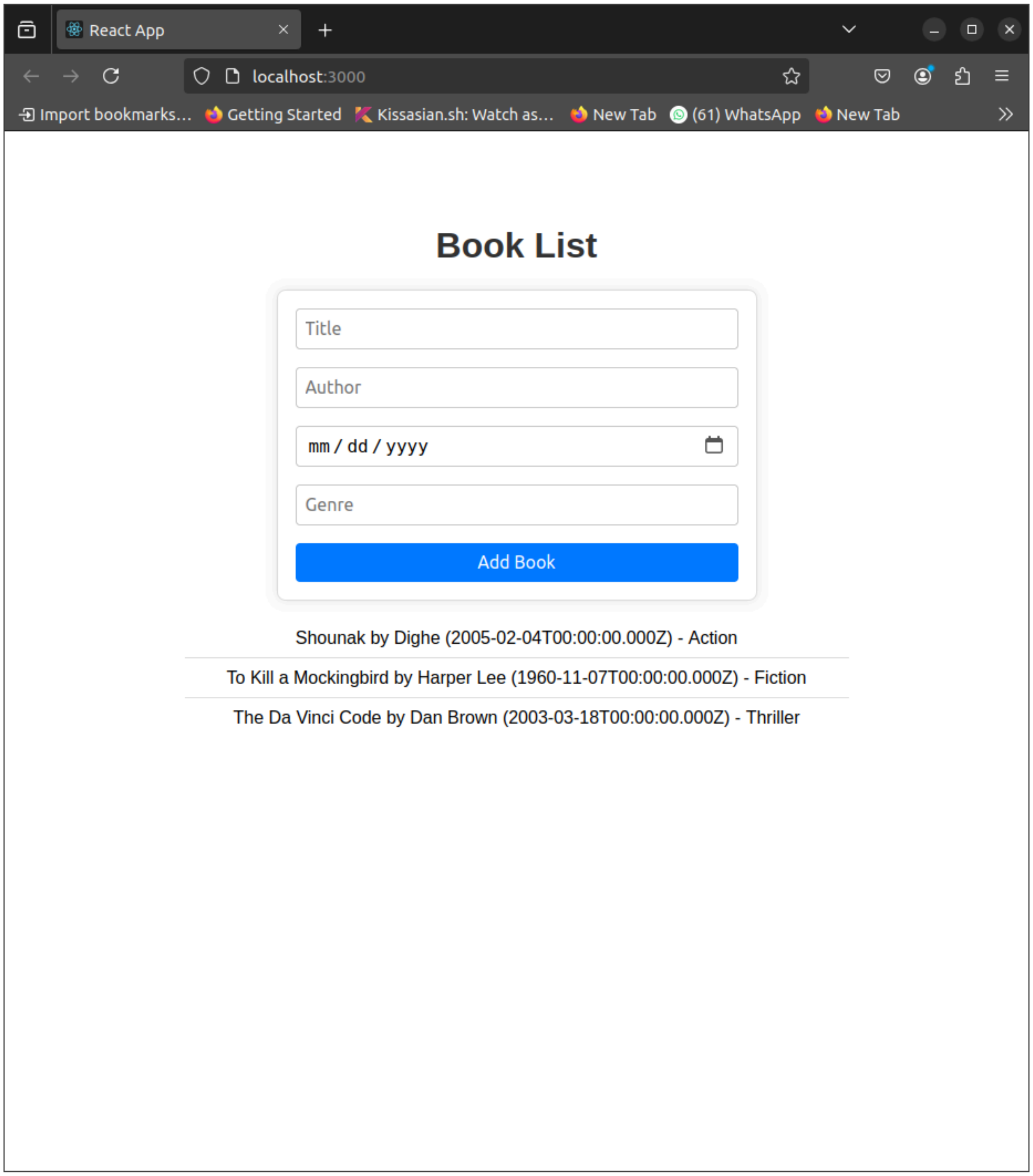
To Kill a Mockingbird by Harper Lee (1960-11-07T00:00:00.000Z) - Fiction


React App
localhost:3000
Import bookmarks... Getting Started Kissasian.sh: Watch as... New Tab (61) WhatsApp New Tab

Book List

The Da Vinci Code
Dan Brown
03 / 18 / 2003
Thriller

Shounak by Dighe (2005-02-04T00:00:00.000Z) - Action
To Kill a Mockingbird by Harper Lee (1960-11-07T00:00:00.000Z) - Fiction



Book List

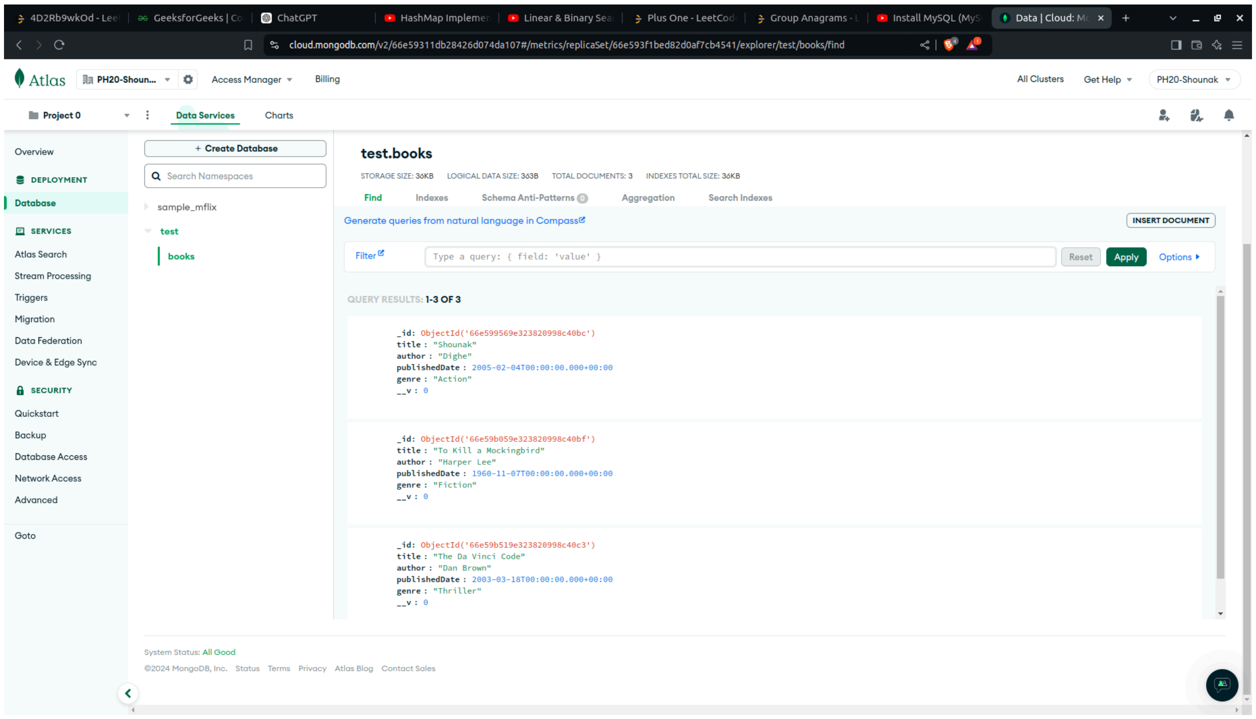
Shounak by Dighe (2005-02-04T00:00:00.000Z) - Action

To Kill a Mockingbird by Harper Lee (1960-11-07T00:00:00.000Z) - Fiction

The Da Vinci Code by Dan Brown (2003-03-18T00:00:00.000Z) - Thriller

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

School of Computer Engineering & Technology
Class: Third Year B.Tech CSE (Semester V)
Course: Full Stack Development



test.books

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 363B TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 36KB

Generate queries from natural language in Compass

Type a query: { field: 'value' }

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('66e599569e323820998c40bc')
title : "Shounak"
author : "Dighe"
publishedDate : 2005-02-04T00:00:00.000+00:00
genre : "Action"
__v : 0

_id: ObjectId('66e59b059e323820998c40bf')
title : "To Kill a Mockingbird"
author : "Harper Lee"
publishedDate : 1960-11-07T00:00:00.000+00:00
genre : "Fiction"
__v : 0

_id: ObjectId('66e59b519e323820998c40c3')
title : "The Da Vinci Code"
author : "Dan Brown"
publishedDate : 2003-03-18T00:00:00.000+00:00
genre : "Thriller"
__v : 0
```

System Status: All Good

©2024 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

server.js code

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');
const bookRoutes = require('./routes/books');

const app = express();

// Middleware
app.use(cors());
app.use(bodyParser.json());
app.use('/books', bookRoutes);

// Connect to MongoDB
```

```
mongoose.connect('mongodb+srv://shounakdighe:<password>@shounakdighe.mnjgh.mongodb.net/?retryWrites=true&w=majority&appName=ShounakDighe', {
   useNewUrlParser: true,
   useUnifiedTopology: true
});

mongoose.connection.on('connected', () => {
   console.log('Connected to MongoDB');
});

mongoose.connection.on('error', (err) => {
   console.log(`MongoDB connection error: ${err}`);
});

// Start the server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => {
   console.log(`Server is running on port ${PORT}`);
});
```